



# BENEMERITA UNIVERSIDAD AUTONOMA DE PUEBLA

FACULTAD DE CIENCIAS DE LA COMPUTACION

Tesina

## **Sistema para la gestión de alumnos de titulación**

Que para obtener el título de Licenciatura en Ingeniería en  
Ciencias de la Computación

Por: Esperanza Alvarado Pineda

Asesor: Dr. Pedro Bello López

Puebla, Pue. Septiembre 2023

## AGRADECIMIENTOS

A Dios Padre por guiar el camino de mi vida. Por darme salud y bendecirme todos los días.

Dedico este proyecto a mis padres Irene Pineda y Enrique Alvarado por sus sacrificios, apoyo incondicional y por haberme brindado la oportunidad de ejercer mi carrera profesional. Por todo lo que me han proporcionado para cumplir mis metas.

A mis compañeras de vida, mis hermanas, por sus enseñanzas y consejos.

Agradezco a mi mejor amigo, mi esposo el Ing. Antonio quien siempre me apoyó durante mis estudios y a la realización de este proyecto, por ser mi compañía y brindarme tanto amor. A mi hijo por ser mi inspiración para realizar mis metas, por ser mi gran amor.

A mis Maestros del diplomado por compartirme sus conocimientos y profesionalismo, en especial al Dr. Pedro Bello por guiarme hasta este momento.

## Contenido

CAPITULO 1. Introducción .....	4
1.1 Planteamiento del problema .....	4
1.2 Antecedentes .....	4
1.3 Justificación .....	4
1.4 Objetivo general.....	5
1.5 Objetivos específicos.....	5
1.6 Estado del arte .....	5
Capítulo 2. Marco Teórico .....	8
2.1 Proceso de titulación.....	8
2.2 Metodología de desarrollo de software.....	8
2.2.1 Proceso Unificado Racional (RUP).....	8
2.3 Lenguaje Unificado de Modelado (UML) .....	10
2.4 Modelo Cliente – Servidor.....	10
2.5 Tecnología de desarrollo .....	11
2.5.1 PHP .....	11
2.5.2 CSS.....	11
2.5.3 HTML 5 .....	12
2.4 Tipo de Bases de Datos .....	12
2.4.1 Modelo Entidad – Relación .....	12
2.4.2 MySQL .....	12
Capítulo 3. Análisis y diseño del sistema.....	13
3.1 Requerimientos del sistema.....	13
3.2 Casos de uso del sistema.....	14
3.3 Especificación de Caso de uso .....	15
3.3.1 Especificación caso de uso: Registro de usuario. ....	15
3.3.2 Especificación de caso de uso: Actualizar estatus de proyecto. ....	16
3.3.3 Especificación de caso de uso: Seguimiento al alumno .....	17
3.4 Diseño del sistema. ....	18
3.4.1 Diagrama de clases.....	18
3.4.2 Diagrama de secuencia .....	19
3.5 Prototipos de interfaces de usuario.....	22

3.5.3 Diseño Interfaz de usuario – Administrador .....	22
3.5.4 Diseño Interfaz de usuario – Docente .....	24
3.5.5 Diseño Interfaz de usuario – Alumno .....	26
3.6 Diseño de la Base de Datos .....	28
CAPITULO 4. Implementación y pruebas del sistema .....	30
4.1 Plataforma de desarrollo.....	30
4.2 Base de datos. ....	30
4.3 Interfaz del Sistema.....	35
4.3.1 Autenticación para acceder al sistema .....	35
4.3.2 Interfaz usuario Administrador. ....	36
4.3.3 Interfaz usuario Docente.....	36
4.3.4 Interfaz usuario Alumno.....	37
4.4 Pruebas.....	37
Conclusión .....	42
Trabajo a futuro .....	42
Bibliografía .....	43
Mesografía .....	43

# CAPITULO 1. Introducción

## 1.1 Planteamiento del problema

La Benemérita Universidad Autónoma de Puebla ofrece diferentes opciones para obtener el título correspondiente a la licenciatura finalizada, algunas de estas opciones son la titulación por medio de tesis que consiste en un trabajo de investigación científica, o bien, por medio de tesina que consiste en realizar un trabajo monográfico para que el estudiante desarrolle los conocimientos obtenidos durante la licenciatura. En ambas de las opciones mencionadas, el proyecto debe ser revisado y aprobado por un asesor de tesis, para que al finalizar el proyecto este sea presentado frente a un jurado.

Cabe mencionar que la Benemérita Universidad Autónoma de Puebla cuenta con un portal virtual de autoservicios para que los docentes y alumnos puedan realizar trámites y gestiones en línea, sin embargo, es un sistema de administración y procesos escolares tanto para alumnos de nuevo ingreso como para egresados.

Es de suma importancia tener un registro de alumnos en procesos de titulación por medio de tesis o tesina para que por medio del seguimiento se conozca el estatus de cada alumno. Se desconoce el método que cada facultad aplica para el registro y seguimiento de los alumnos en proceso de titulación por tal motivo se propone desarrollar un sistema para la gestión de alumnos únicamente en proceso de titulación por medio de tesis o tesina.

## 1.2 Antecedentes

Hoy en día en la Facultad de Ciencias de la Computación no cuenta con un sistema único para la gestión, control y seguimiento de alumnos en proceso de titulación por medio de tesis o tesina. Por lo anterior, en este proyecto se desarrollará un Sistema para la gestión de alumnos de titulación por dicho medio de titulación.

Con el sistema para la gestión de alumnos en proceso de titulación la Facultad de Ciencias de la Computación facilitara el seguimiento y revisión de proyectos para los docentes como para los alumnos.

## 1.3 Justificación

Las tecnologías de la información son empleadas para desarrollar herramientas educativas, como gestores de aprendizaje, sistemas de administración escolar entre otras. La mayoría de las universidades y planteles educativos ya cuentan con un sistema que les permite llevar el control de la historia académica por cada estudiante, sin embargo, los sistemas que ya existen tienen un costo o son plantillas que se personalizan para cada escuela.

En el mercado no existe un sistema gestor únicamente de egresados en proceso de titulación por medio de tesis y para no pagar licencias o suscripciones se desarrollará un sistema para la gestión de alumnos de titulación con software libre para garantizar el control de alumnos tesistas sin que este genere un costo adicional a la universidad ni al estudiante.

## 1.4 Objetivo general

El objetivo general de este proyecto es implementar un sistema para la gestión de alumnos en proceso de titulación mediante tesis o tesina, facilitando a los alumnos y docentes el acceso a la información y estatus del proyecto.

## 1.5 Objetivos específicos

- Realizar la especificación de requerimientos.
- En base a los requerimientos diseñar la base de datos relacional.
- Diseñar la interfaz del sistema usando CSS y HTML.
- Garantizar el almacenamiento de los datos del alumno, así como descripción, estatus y avance de su proyecto de tesis o tesina.
- Generar reportes del avance del proyecto.

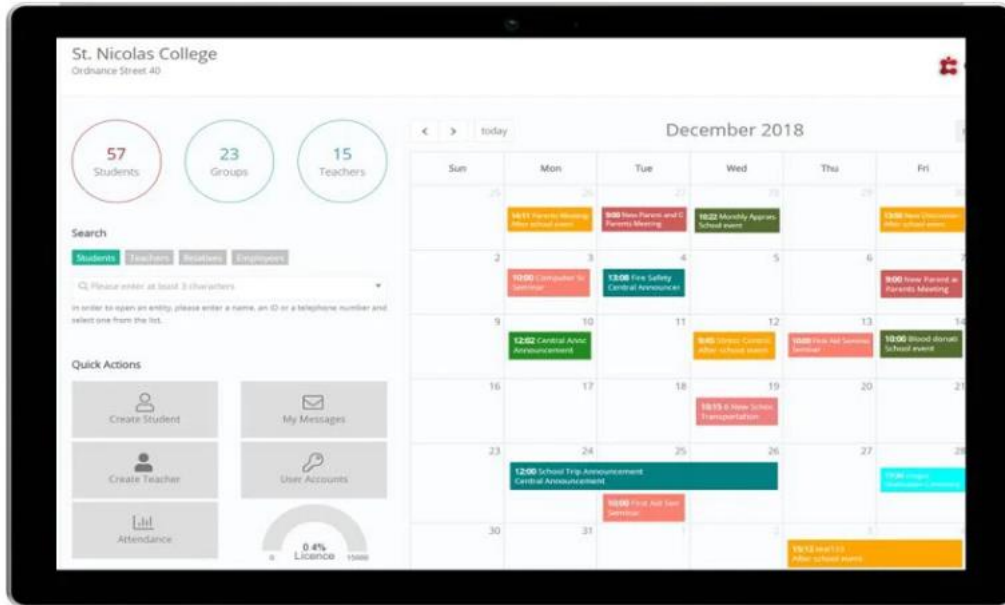
## 1.6 Estado del arte

En cada Universidad existen diferentes opciones de titulación dependiendo el plan de estudios vigente, sin embargo, una de las opciones que se aplica en cualquier universidad es la titulación mediante tesis o tesina. Para la administración y seguimiento de los alumnos tesistas cada escuela lleva su control interno apoyándose de herramientas como sistemas de gestión educativo.

De acuerdo con la investigación realizada sobre el tema de sistemas para la gestión de alumnos de titulación se han encontrado sistemas de gestión escolar con funciones similares a las que se desarrollaran en este proyecto, los cuales se mencionan a continuación:

- **Classter**  
<https://www.classter.com/>

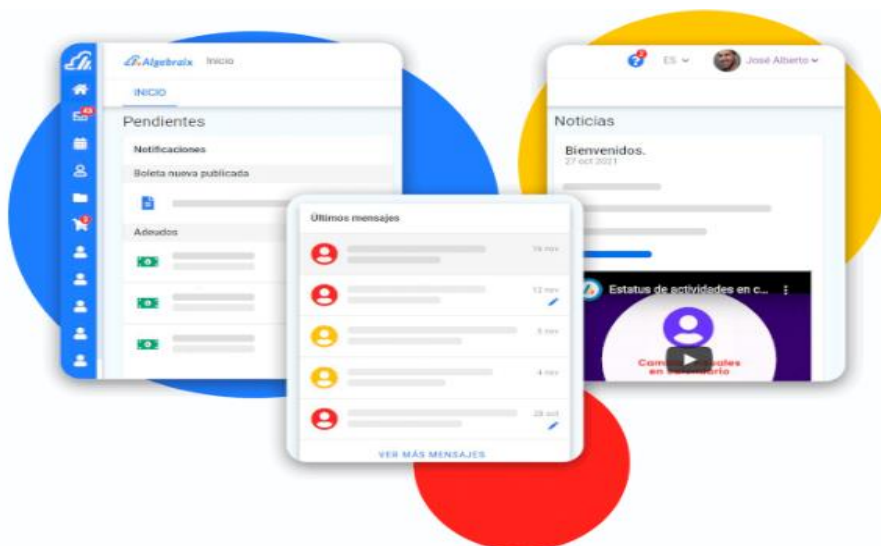
Ayuda a las organizaciones educativas a optimizar las operaciones administrativas y digitalizar los procesos académicos.



Es un sistema muy completo que facilita la gestión administrativa y procesos académicos del alumno, sin embargo, no está enfocado al proceso de titulación, por lo que no se adecua a nuestro requerimiento principal de la gestión de alumnos solo en proceso de titulación [13].

- **Algebraix**  
<https://www.algebraix.com>

Todas las herramientas que tu escuela necesita en un mismo lugar. Ofrece desarrollo constante, implementación guiada, atención Premium.



Es la plataforma que le brinda a la comunidad escolar las herramientas académicas, financieras y de comunicación para mejorar la interacción entre Padres de Familia, Estudiantes, Maestros, Personal Administrativo y Directivos. Esta plataforma ofrece un servicio muy completo, sin embargo, el costo de la renta se adapta dependiendo la cantidad de usuarios y nivel educativo de cada institución [14].

A pesar de que actualmente en el mercado existen diversos sistemas de gestión escolar, algunos con costo de licencia, suscripción o membresía, o bien, gratuitos, no hay un sistema que cumpla únicamente con los requerimientos del proyecto propuesto en esta tesina. Los cuales se describen más adelante.

## Capítulo 2. Marco Teórico

Como en todo proyecto es importante definir los conceptos teóricos metodológicos que se aplican antes y durante el desarrollo del proyecto, a continuación, se describen los conceptos básicos para el desarrollo de este proyecto.

### 2.1 Proceso de titulación

Cuando un alumno de educación superior culmina sus estudios inicia el proceso de titulación, que corresponde a la última etapa de la universidad, consiste en demostrar que el estudiante ha cursado íntegramente los estudios de nivel superior, es decir, la licenciatura.

El título profesional es expedido por la universidad donde el egresado cursó los estudios de licenciatura, el trámite se realiza directamente con la dirección escolar y los requisitos y proceso depende de las políticas de cada universidad, así como el costo y tiempos de entrega.

### 2.2 Metodología de desarrollo de software

Una metodología de desarrollo de software es un modo sistemático de realizar, gestionar y administrar un proyecto. La ingeniería de software es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, la operación y el mantenimiento al software [4].

La mayor parte de los modelos de procesos de software se basan en uno de los tres modelos o paradigmas de desarrollo de software [5]:

- El enfoque en cascada
- Desarrollo iterativo
- Ingeniería de software basada en componentes

Para el desarrollo de este proyecto de tesina se aplicará la metodología RUP (Rational Unified Process) basada en UML, ya que proporciona un enfoque disciplinado para asignar tareas durante la organización del desarrollo. Su objetivo es garantizar que el desarrollo del software cumpla con todos los requerimientos del usuario dentro de un presupuesto y tiempos establecidos.

#### 2.2.1 Proceso Unificado Racional (RUP)

El Proceso Unificado Racional es un marco de trabajo extensible que puede ser adaptado a organizaciones o proyectos específicos. Se puede describir en función de dos dimensiones [1]:

- Dimensión temporal: se expresa en términos de ciclos, fases, iteraciones e hitos.
- Dimensión estática: se expresa en términos de actividades, productos intermedios, perfiles de trabajo o roles y flujos de trabajo.

Atendiendo a la dimensión temporal, el ciclo de vida se divide en ciclos entendiendo a estos como vueltas alrededor de una espiral, y estas, a su vez, como los periodos de tiempo en los que se trabaja sobre una versión completa del sistema. En la figura 1 se observa que cada ciclo se compone de cuatro fases que se realizan secuencialmente:

- Inicio: definir el alcance del proyecto
- Elaboración: definición, análisis, diseño
- Construcción: implementación
- Transición: fin del proyecto y puesta en producción

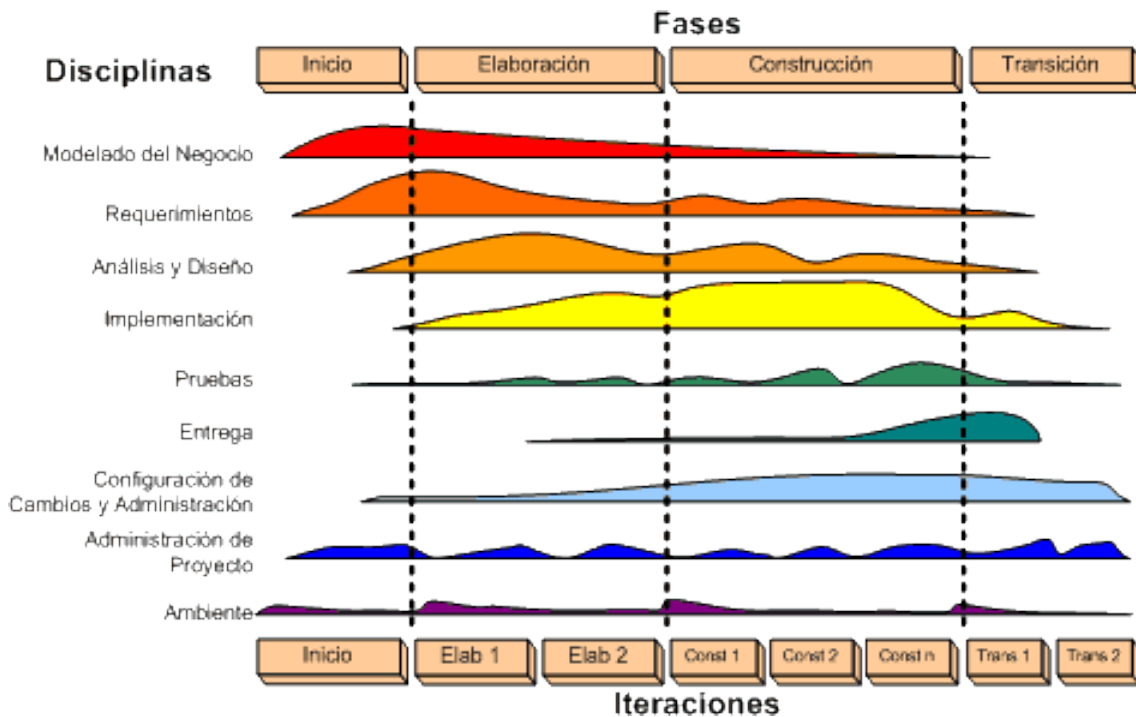


Figura 1 Dimensiones y fases del RUP

Cada una de las fases de RUP puede descomponerse en iteraciones. Una iteración es un periodo de tiempo en el que se realiza un conjunto completo de actividades de desarrollo. El sistema, por lo tanto, se va desarrollando incrementalmente de iteración en iteración. Una nueva iteración produce una nueva versión que proporciona al software mayor funcionalidad y es más refinada [2].

Atendiendo a la dimensión estática, el proceso RUP establece cuatro elementos de modelado principales [2]:

- Perfiles o papeles de trabajo
- Actividad
- Producto intermedio

- Flujos de trabajo
  - Flujos de ingeniería (actividades secuenciales)
    - Modelado del negocio
    - Requerimientos
    - Análisis y diseño
    - Implementación
    - Pruebas
    - Implantación
  - Flujos de apoyo (actividades de gestión paralelas)
    - Gestión de la configuración
    - Gestión de proyecto
    - Entorno

Otra de las características esenciales que definen al RUP es que es un proceso dirigido por casos de uso. Dado que utiliza casos de uso para el inicio, desenvolvimiento y desarrollo de las disciplinas con los artefactos, roles y actividades necesarias. Los casos de uso son la base para la implementación de las fases.

### 2.3 Lenguaje Unificado de Modelado (UML)

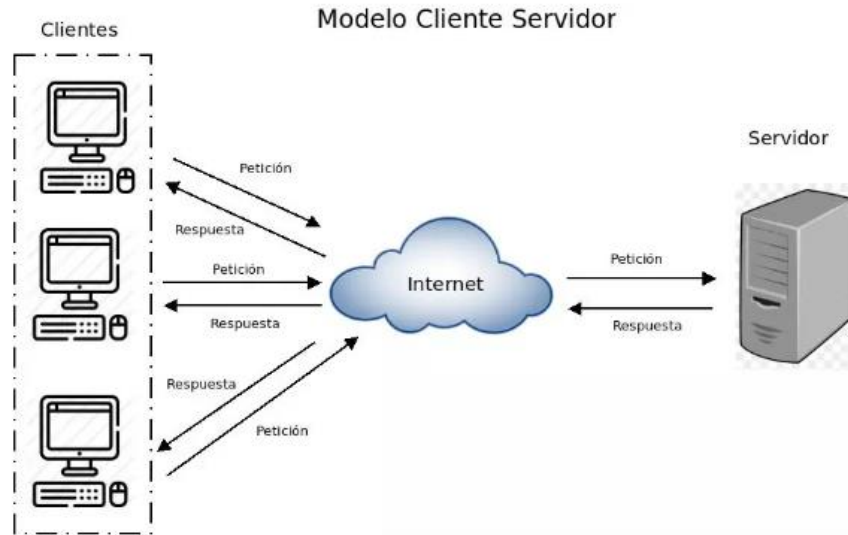
Dado que en la metodología de Proceso Unificado Racional los casos de uso son la base para la implementación de las fases, el diseño de diagramas, modelos visuales de objetos, estados y procesos de un sistema son elaborados con el uso de lenguaje UML.

UML es un lenguaje para hacer modelos y es independiente de los métodos de análisis y diseño [15]. El lenguaje UML tiene una notación gráfica muy expresiva que permite representar en mayor o menor medida todas las fases de un proyecto. Por un lado, el lenguaje de modelado puede servir de modelo para un proyecto y garantizar así una arquitectura de información estructurada; por el otro, ayuda a los desarrolladores a presentar la descripción del sistema de una manera que sea comprensible para quienes están fuera del campo. UML se utiliza principalmente en el desarrollo de software orientado a objetos.

### 2.4 Modelo Cliente – Servidor

El modelo arquitectónico cliente – servidor es un modelo de sistema en el que dicho sistema se organiza como un conjunto de servicios y servidores asociados, más unos clientes que acceden y usan los servicios [16]. Los principales componentes de este modelo son:

- Servidor: un servidor hace referencias a un proveedor de servicios.
- Cliente: o conjunto de clientes que solicitan a los servicios ofrecidos por el servidor.
- Red: medio por el cual los clientes acceden a los servicios



*Figura 2 Esquema del modelo Cliente - Servidor*

La principal importancia de este modelo es que permite conectar a varios clientes a los servicios que provee un servidor. Por otra parte, este modelo garantiza cierto nivel de seguridad ya que el servidor por lo general está protegido por un firewall o subredes que impiden que los atacantes puedan acceder a la base de datos [16].

## 2.5 Tecnología de desarrollo

El sistema para la gestión de alumnos en titulación deberá ser desarrollado utilizando HTML 5, CSS Y PHP.

### 2.5.1 PHP

PHP es el lenguaje de programación diseñado para la creación de páginas Web dinámicas, más eficiente, más rápido y fácil de aprender. Entre sus ventajas no está sólo que pueda ejecutarse en casi cualquier sistema operativo, sino que, a diferencia de otros lenguajes de programación, le permite intercambiar su trabajo de una plataforma a otra con muy pocas o ninguna modificación [6] [7].

PHP es un lenguaje de programación abierto, es decir, no hay que pagar una licencia para poder adquirirlo, ofrece una amplia variedad de características avanzadas para los desarrolladores web. Este lenguaje de programación se ejecuta del lado del servidor.

### 2.5.2 CSS

CSS. Hojas de estilo en cascada, el lenguaje CSS está basado en una serie de reglas que rigen el estilo de los elementos en los documentos estructurados y que forman la sintaxis de las hojas de estilo. Cada regla consiste en un selector y una declaración, esta última va entre corchetes y consiste en una propiedad o atributo y un valor separado por dos puntos. Se les denomina hojas de estilo en cascada porque puedes tener varias hojas y una de ellas con las propiedades heredadas de otras.

CSS funciona como complemento a la información que forma parte de un sitio web. Mientras que HTML incluye todos los datos y complementos, el código CSS se encarga de darles formato y presentarlos visualmente a través de un navegador. [17]

### 2.5.3 HTML 5

HTML 5 no es un lenguaje de programación, sin embargo, es considerado un lenguaje de marcas de hipertexto para la estructura de un sitio web. Sin HTML no hay página web. La sintaxis es fácil y clara para todos los elementos que podemos agregar en nuestro sitio web [10].

## 2.4 Tipo de Bases de Datos

Las bases de datos se usan para satisfacer las necesidades de información de muchas organizaciones e individuos en una variedad de áreas. El estudio de la teoría, diseño y gestión de bases de datos permite maximizar sus beneficios potenciales. La gestión de datos implica tanto la definición de estructuras para almacenar la información como la provisión de mecanismos de manipulación de la información [9].

Todo acceso a la base de datos es a través del sistema de gestión de base de datos (DBMS), un paquete de software que configura estructuras de almacenamiento. El DBMS proporciona una interfaz de programa y una interfaz de usuario para consultas interactivas que se expresan en el lenguaje del DBMS particular [18].

### 2.4.1 Modelo Entidad – Relación

Es un modelo conceptual de alto nivel de abstracción, está basado en una percepción del mundo real que consta de un conjunto de objetos básicos llamados entidades y relaciones entre estos objetos. Representa gráficamente y de manera lógica toda la información y como los datos se relacionan entre sí [12].

El modelo Entidad – Relación tiene tres elementos principales:

- Entidades: cada uno de los componentes del proceso de negocio.
- Atributos: propiedades de las entidades.
- Relaciones: vínculos entre parejas de entidades.

### 2.4.2 MySQL

Para almacenar y trabajar con los datos el sistema para la gestión de alumnos en titulación usará el gestor de base de datos MySQL. MySQL es un gestor de bases de datos relacional, desarrolla alta velocidad en la búsqueda de datos e información, además de que es un software que no genera un costo. Es compatible con diferentes sistemas operativos, por lo tanto, no limita al desarrollador en cuanto al sistema operativo base [11].

## Capítulo 3. Análisis y diseño del sistema.

De acuerdo con lo mencionado en el capítulo anterior, para la implementación del presente trabajo se utiliza la metodología de software del Proceso Unificado (RUP), por lo que a continuación se procede al análisis y diseño del sistema, describiendo los requerimientos funcionales y no funcionales, así como el diseño del sistema representado por diagramas.

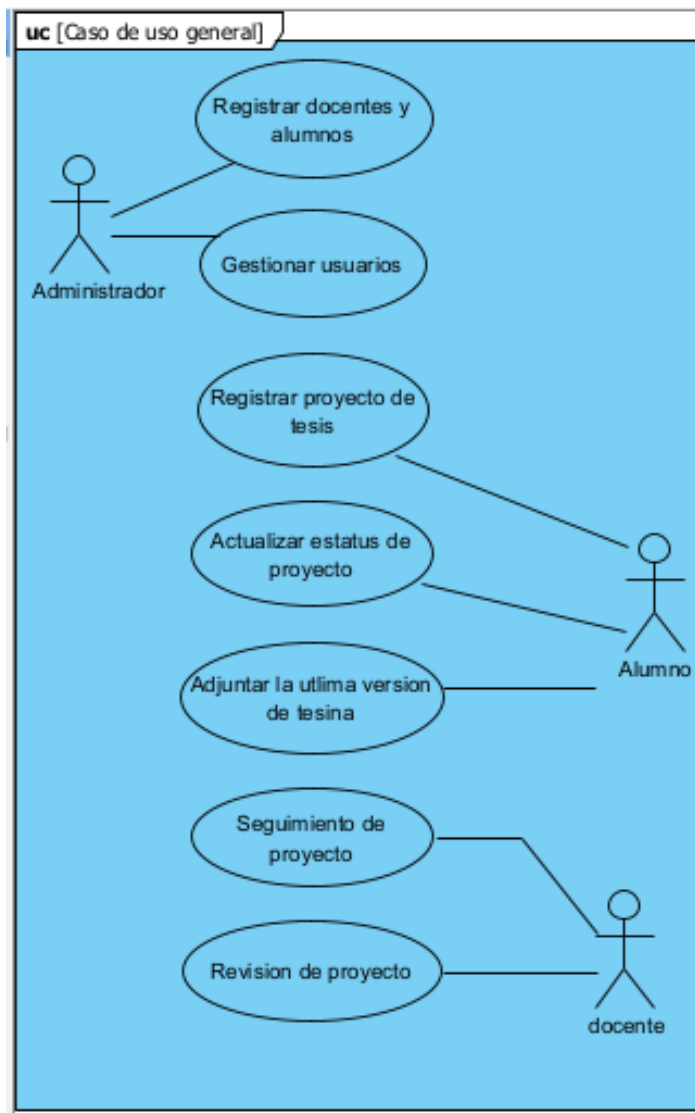
### 3.1 Requerimientos del sistema

N° Req.	Requerimiento	Descripción
<b>RF1</b>	Iniciar sesión	El sistema validará el nombre de usuario y contraseña cuando se detecte un inicio de sesión
<b>RF2</b>	Control de acceso por tipo de rol	El sistema debe contemplar 3 tipos de usuarios: administrador, docente y alumno
<b>RF3</b>	Usuario administrador	El sistema permitirá al usuario de tipo administrador agregar, consultar, actualizar y eliminar los registros de docentes y alumnos.
<b>RF4</b>	Usuario docente	El sistema permitirá al usuario de tipo docente la consulta de los alumnos tesis asignados, así como los datos de su proyecto de tesis.
<b>RF5</b>	Usuario alumno	El sistema permitirá al usuario de tipo alumno actualizar los datos de su trabajo de tesis, en específico el estatus y almacenar la última versión del documento de tesina.
<b>RF6</b>	Sistema	El sistema como tal debe tener una base de datos para el control de sesiones y almacenamiento de la información
<b>RN1</b>	Usabilidad 1	El sistema web puede ser consultado desde cualquier navegador
<b>RN2</b>	Rendimiento	Se espera que el tiempo de una consulta en la base de datos no exceda los 20 segundos.
<b>RNF3</b>	Usabilidad 2	La interfaz de usuario del sistema debe ser amigable e intuitiva de usar.
<b>RNF4</b>	Adjuntar documento	El sistema debe permitir solo archivos en formato PDF

### 3.2 Casos de uso del sistema

En base a la descripción del sistema especificada en los requerimientos de usuario y sistema, se diseña el modelo de casos de uso identificando los actores principales.

- Caso de uso general del sistema



*Figura 3 Diagrama de caso de uso general*

- Caso de uso específico – Gestionar usuarios

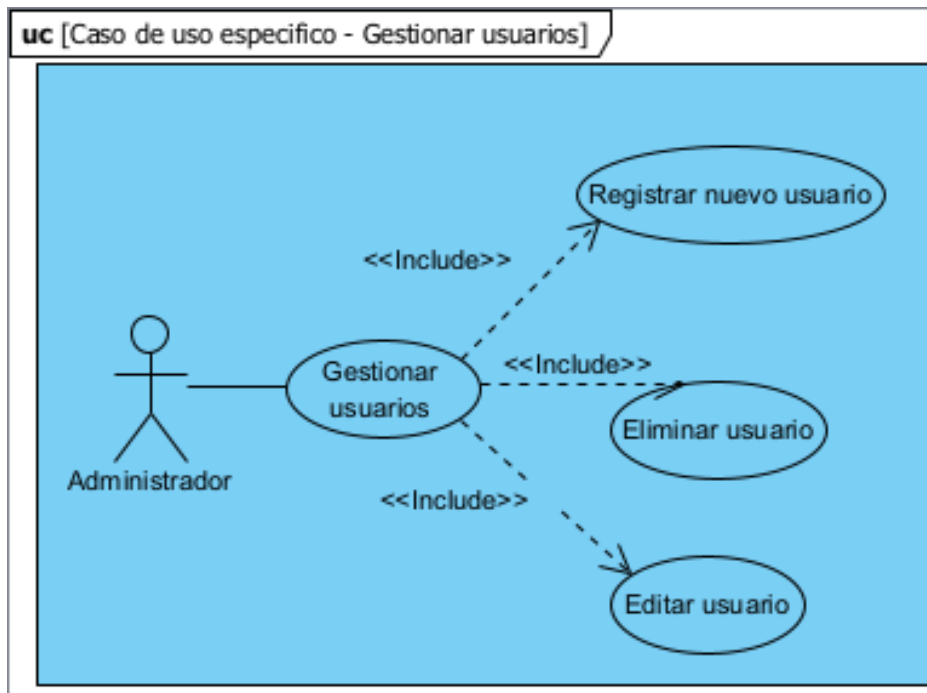


Figura 4 Diagrama de caso de uso específico – Gestionar usuarios

### 3.3 Especificación de Caso de uso

#### 4.3.1 Especificación caso de uso: Registro de usuario.

Caso de uso	Registro de usuario en el sistema de gestión de alumnos de titulación
<b>Actor:</b>	Administrador
<b>Precondición:</b>	El usuario debe iniciar sesión como administrador,
<b>Descripción:</b>	Un alumno que está por iniciar su proyecto de tesina o un docente asignado como asesor, solicita una cuenta de acceso al administrador.
<b>Flujo básico:</b>	<p>Para registrar un nuevo usuario se debe seguir el siguiente flujo</p> <ol style="list-style-type: none"> <li>1) El administrador debe identificar el tipo de usuario a registrar</li> <li>2) El sistema muestra formulario de registro</li> <li>3) El administrador captura los datos solicitados</li> <li>4) El sistema valida que no haya un registro igual</li> <li>5) El sistema almacena la información en la base de datos</li> </ol>
<b>Postcondición</b>	Los datos del usuario han sido registrados con éxito. Ya puede iniciar sesión.

<b>Excepciones</b>	Si el sistema no registra la información en la base de datos: <ul style="list-style-type: none"> <li>1) El sistema identifica el tipo de error</li> <li>2) El sistema informa al usuario mediante un mensaje en pantalla el error: error en la conexión con BD, usuario existente, información proporcionada errónea.</li> <li>3) Restaurar caso de uso.</li> </ul>
<b>Comentarios</b>	Los datos para el inicio de sesión de la cuenta creada son compartidos por correo,

### 3.3.2 Especificación de caso de uso: Actualizar estatus de proyecto.

Caso de uso      Actualizar estatus de proyecto de tesina	
<b>Actor:</b>	Alumno
<b>Precondición:</b>	El alumno debe estar previamente registrado en el sistema, así como la información de su proyecto
<b>Descripción:</b>	El asesor del alumno requiere conocer el estatus del proyecto directamente en línea por lo que el alumno debe estar actualizando dicho estatus en el sistema.
<b>Flujo básico:</b>	Para actualizar el estatus del proyecto se debe seguir el siguiente flujo <ul style="list-style-type: none"> <li>1) El alumno inicia sesión en el sistema web</li> <li>2) Selecciona la opción 'Agregar avances' del menú principal</li> <li>3) Clic en el botón actualizar</li> <li>4) El sistema muestra el formulario para actualizar la información deseada</li> <li>5) El alumno modifica la información de su proyecto</li> <li>6) El sistema valida la nueva información y la almacena en la base de datos.</li> </ul>
<b>Postcondición</b>	Se ha actualizado la información de proyecto de tesina
<b>Excepciones</b>	Si el sistema no actualiza la información en la base de datos: <ul style="list-style-type: none"> <li>1) El sistema identifica el tipo de error</li> <li>2) El sistema informa al usuario mediante un mensaje en pantalla el error: error en la conexión con BD, error de archivo.</li> <li>3) Restaurar caso de uso.</li> </ul>
<b>Comentarios</b>	El alumno debe estar actualizando el estatus frecuentemente para que su asesor realice el seguimiento correspondiente.

### 3.3.3 Especificación de caso de uso: Seguimiento al alumno

Caso de uso	
Seguimiento al alumno	
<b>Actor:</b>	Docente
<b>Precondición:</b>	El docente debe estar previamente registrado en el sistema y además tener asignado al menos un tesista.
<b>Descripción:</b>	Como su rol lo indica el docente debe asesorar a su alumno tesista asignado por lo que debe conocer los datos del alumno, así como la información de su proyecto de tesina.
<b>Flujo básico:</b>	<p>Para que el docente pueda realizar el seguimiento a su alumno se debe seguir el siguiente flujo</p> <ol style="list-style-type: none"><li>1) El docente inicia sesión en el sistema web</li><li>2) Selecciona la opción 'Lista de alumnos'</li><li>3) El sistema muestra los datos actualizados del proyecto del alumno seleccionado</li><li>4) El docente realiza la revisión</li><li>5) Para agregar una retroalimentación selecciona la opción 'Agregar comentarios'</li><li>6) El sistema muestra el cuadro de texto</li><li>7) El docente captura sus comentarios</li><li>8) El sistema almacena los comentarios en la base de datos.</li></ol>
<b>Postcondición</b>	Se realiza retroalimentación de proyecto por cada alumno seleccionado
<b>Excepciones</b>	<p>Si el sistema no muestra la información de los alumnos o del proyecto:</p> <ol style="list-style-type: none"><li>9) El sistema identifica el tipo de error</li><li>10) El sistema informa al usuario mediante un mensaje en pantalla el error: error en la conexión con BD, error de selección.</li><li>11) Restaurar caso de uso.</li></ol>
<b>Comentarios</b>	En caso de que el estatus del alumno sea baja, el administrador debe eliminar el registro del alumno, así como los del proyecto registrado.

### 3.4 Diseño del sistema.

Continuando con el diseño del sistema, se realiza el modelado de diagramas con UML para visualizar, especificar y documentar los actores, operaciones y componentes del presente proyecto.

#### 4.4.1 Diagrama de clases

En el diagrama de clases (figura 5) se muestran los atributos y operaciones de cada clase, así mismo el tipo de relaciones existentes entre ellas.

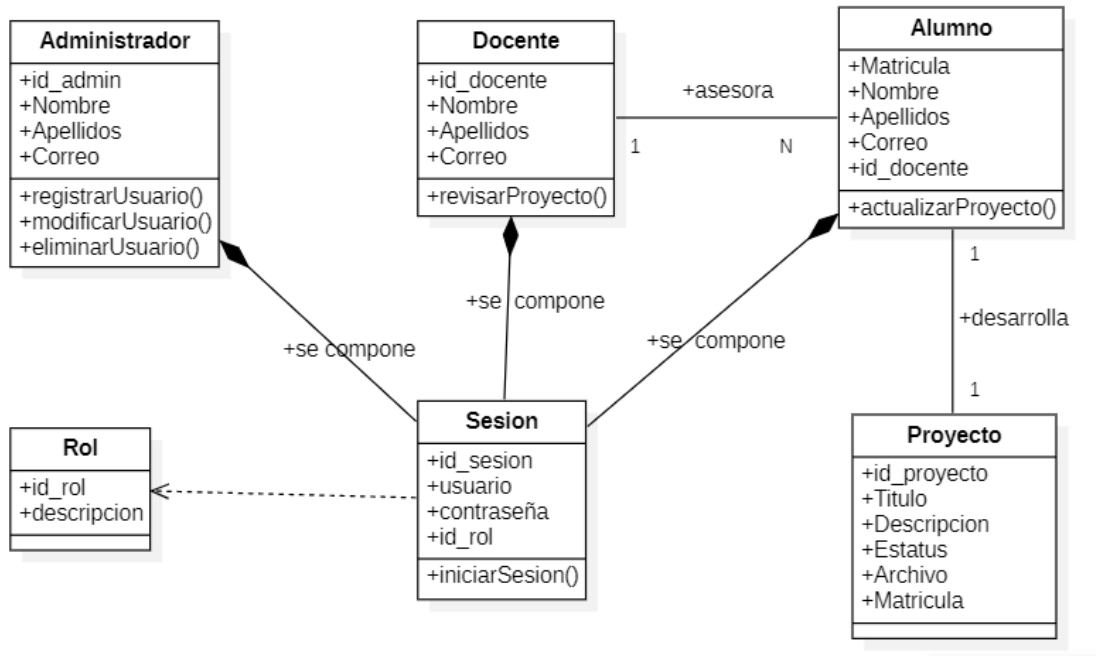


Figura 5 Diagrama de clases

### 3.4.2 Diagrama de secuencia

Un diagrama de secuencia muestra la interacción entre los componentes principales del sistema, en el cual se especifican los módulos o clases que forman parte del sistema y las llamadas que se realizan para ejecutar una tarea determinada, por esta razón permite observar la perspectiva cronológica de las interacciones. A continuación, se muestran los diagramas de secuencia correspondientes a los casos de uso expuestos en el punto anterior.

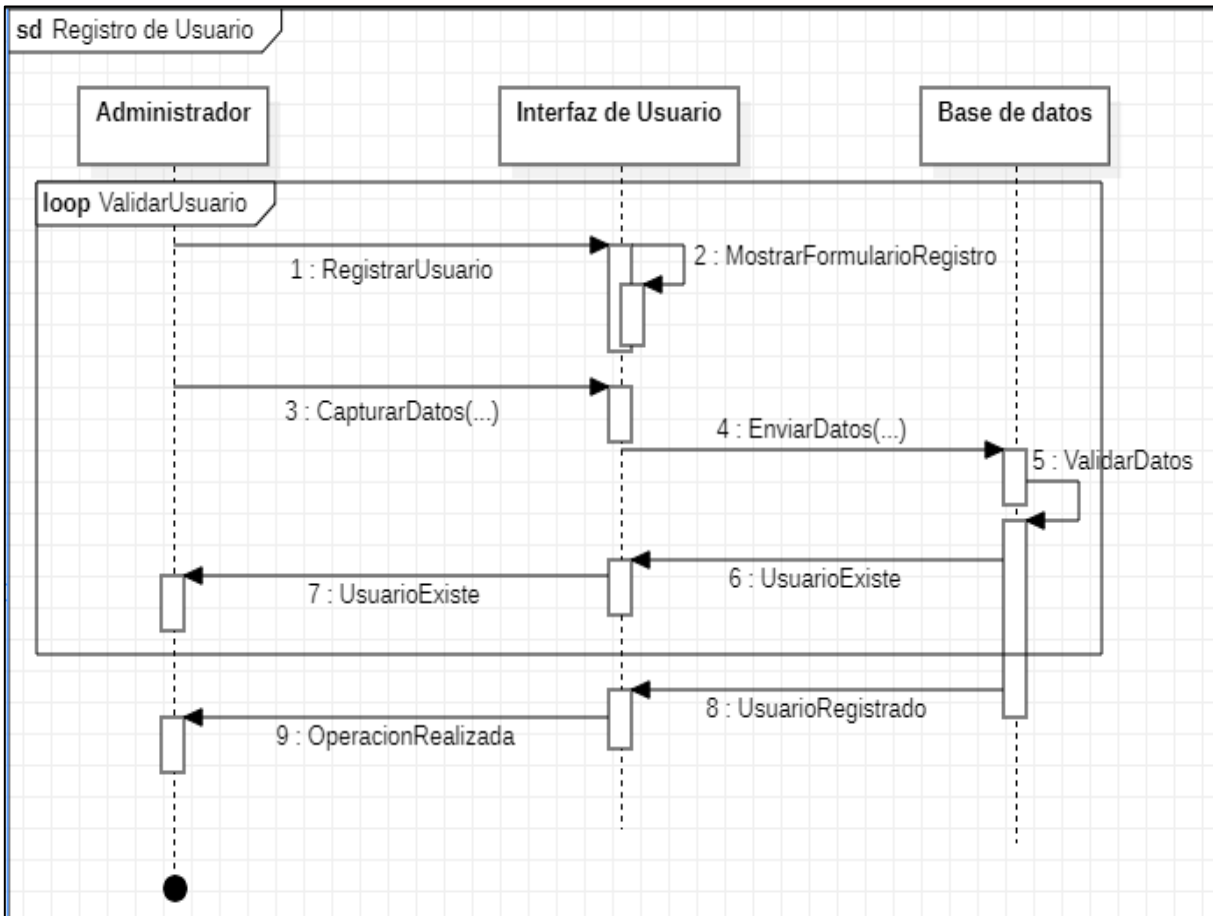


Figura 6 Diagrama de Secuencia - Caso de Uso Registrar usuario

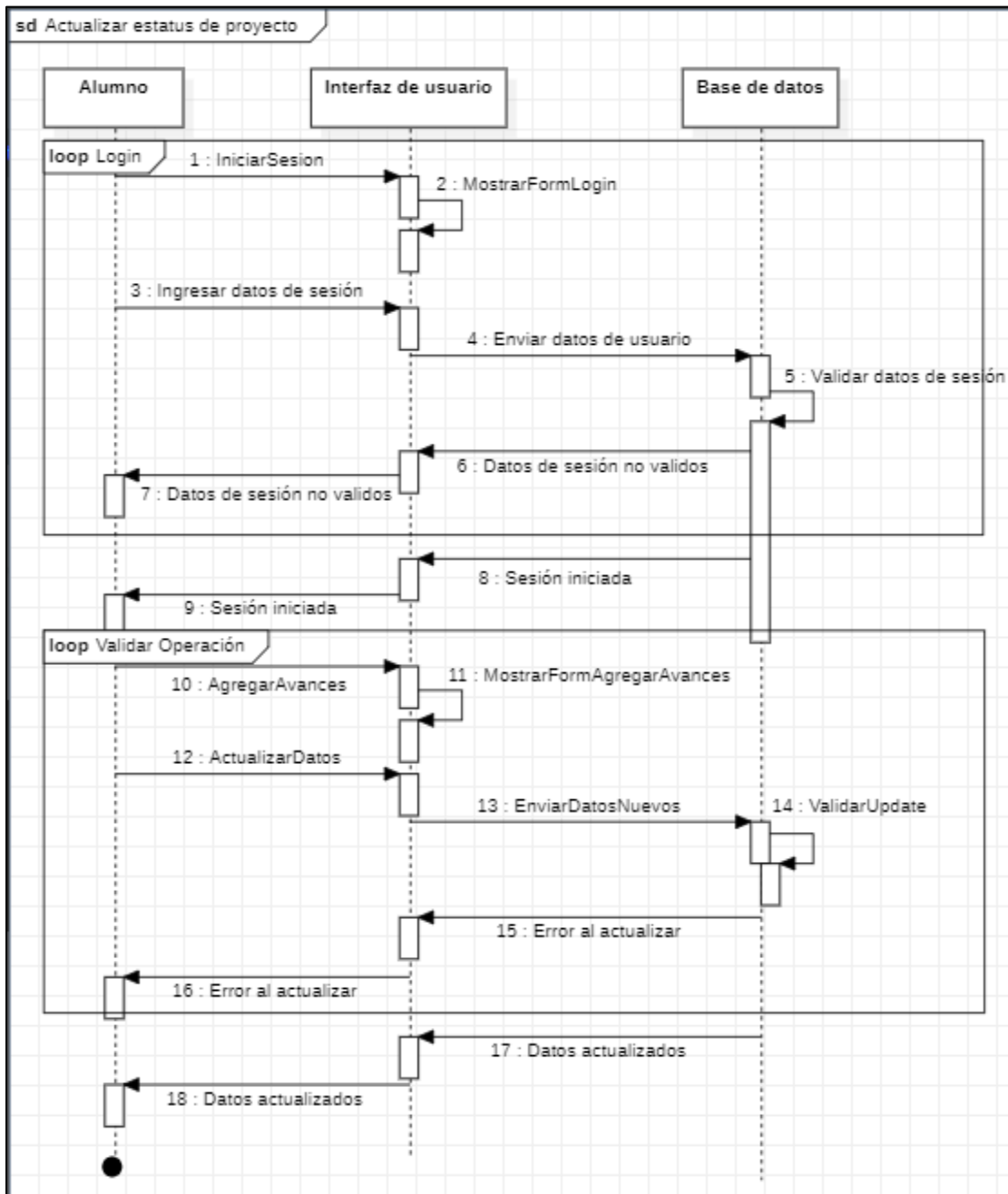


Figura 7 Diagrama de Secuencia - Caso de Uso Actualizar estatus de proyecto

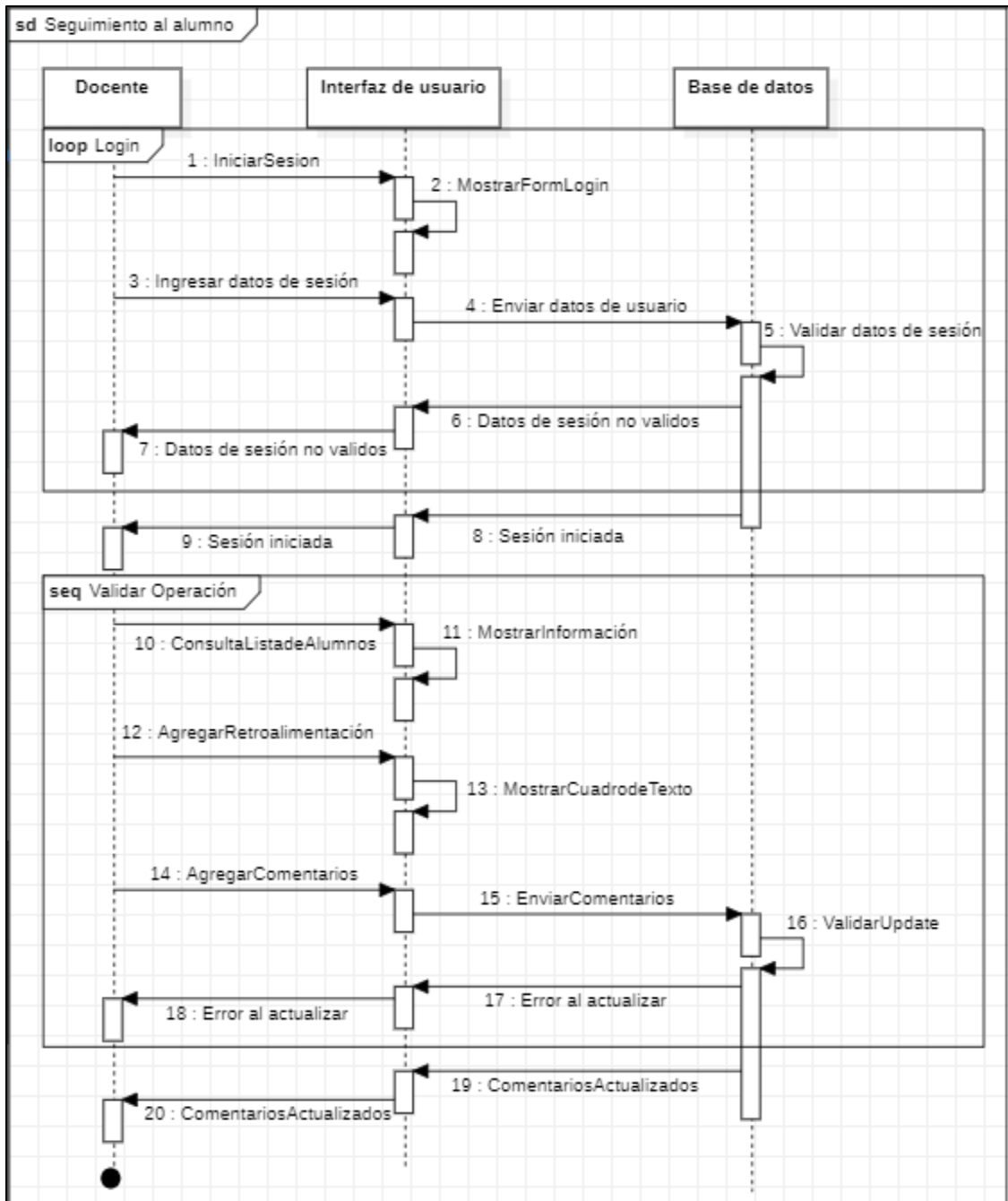


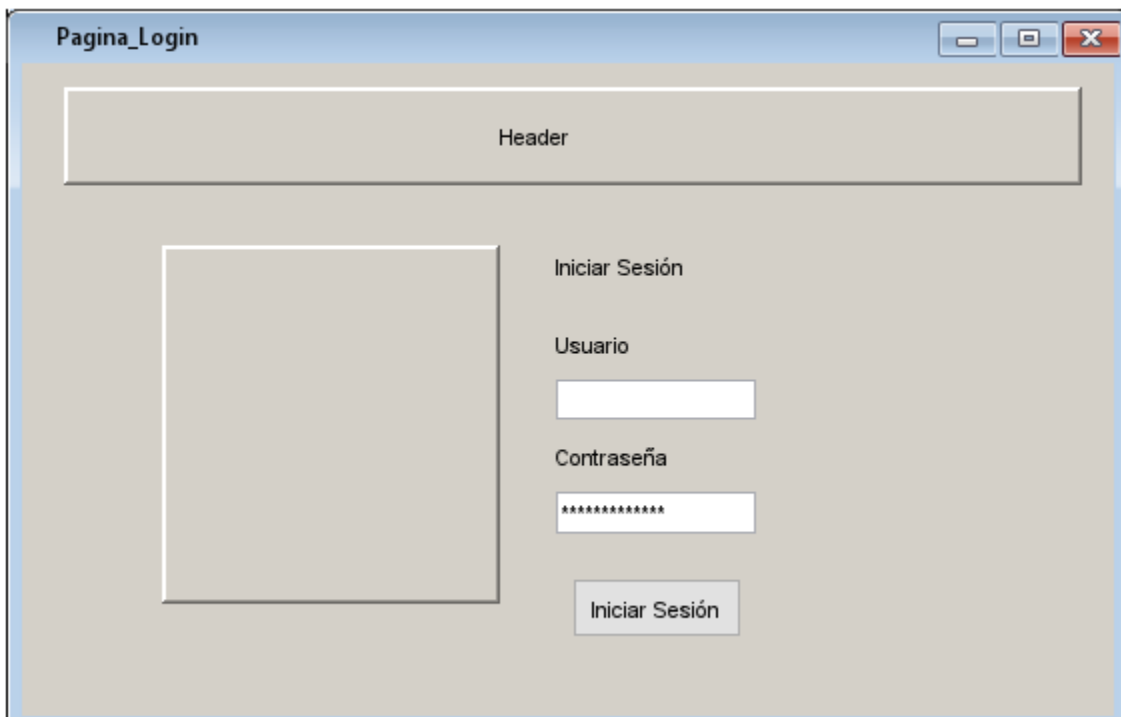
Figura 8 Diagrama de Secuencia - Caso de Uso Seguimiento al alumno

### 3.5 Prototipos de interfaces de usuario

A continuación, se presenta el diseño de prototipos de las principales interfaces de usuario agregando una breve descripción del funcionamiento de cada una de ellas.

→ Prototipo interfaz inicio de sesión en el sistema

La pantalla de inicio de sesión para los usuarios previamente registrados estará estructurada como se muestra en la figura 9, los usuarios deberán ingresar su nombre de usuario y contraseña, los cuales serán validados por el sistema para identificar el tipo de usuario que desea iniciar sesión.



*Figura 9 Elaboración propia del prototipo - Inicio de sesión*

#### 3.5.3 Diseño Interfaz de usuario – Administrador

Si el usuario inicia sesión como administrador se mostrará la página de inicio (figura 10), el administrador podrá navegar en un menú con las opciones: Inicio, Docentes, Alumnos y Cerrar sesión. Donde la última opción, claramente es para salir del sistema.

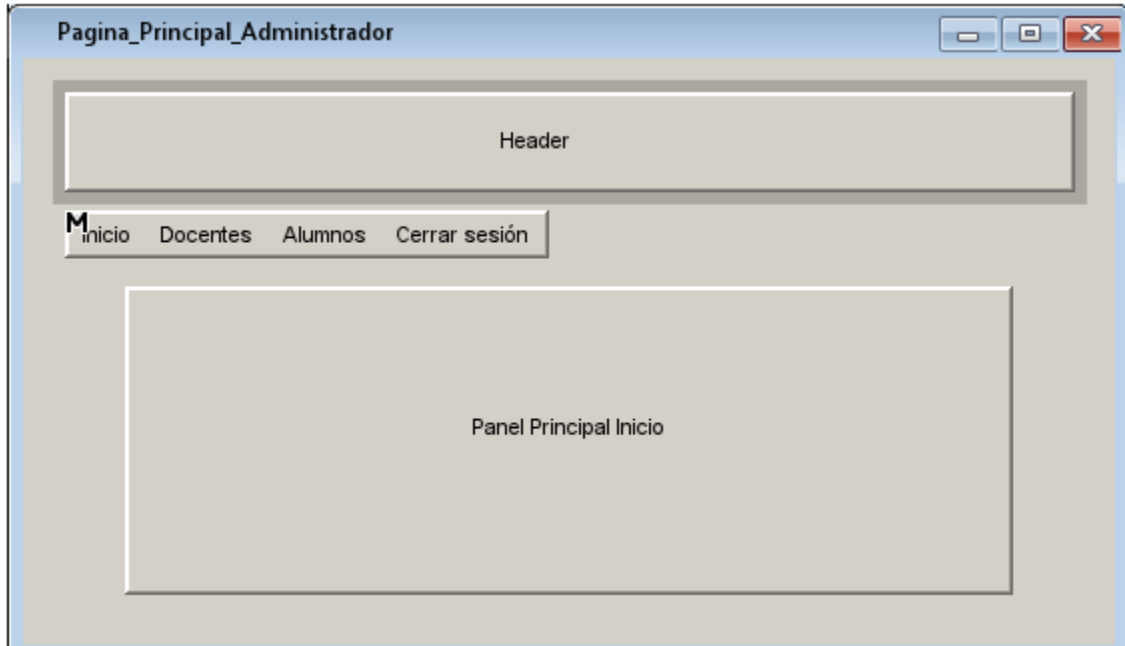


Figura 10 Elaboración propia del prototipo - Página principal administrador

En la opción 'Docentes' (figura 11) el usuario de tipo administrador podrá visualizar todos los docentes registrados en el sistema, para registrar un docente nuevo debe dar clic en el botón agregar docente.

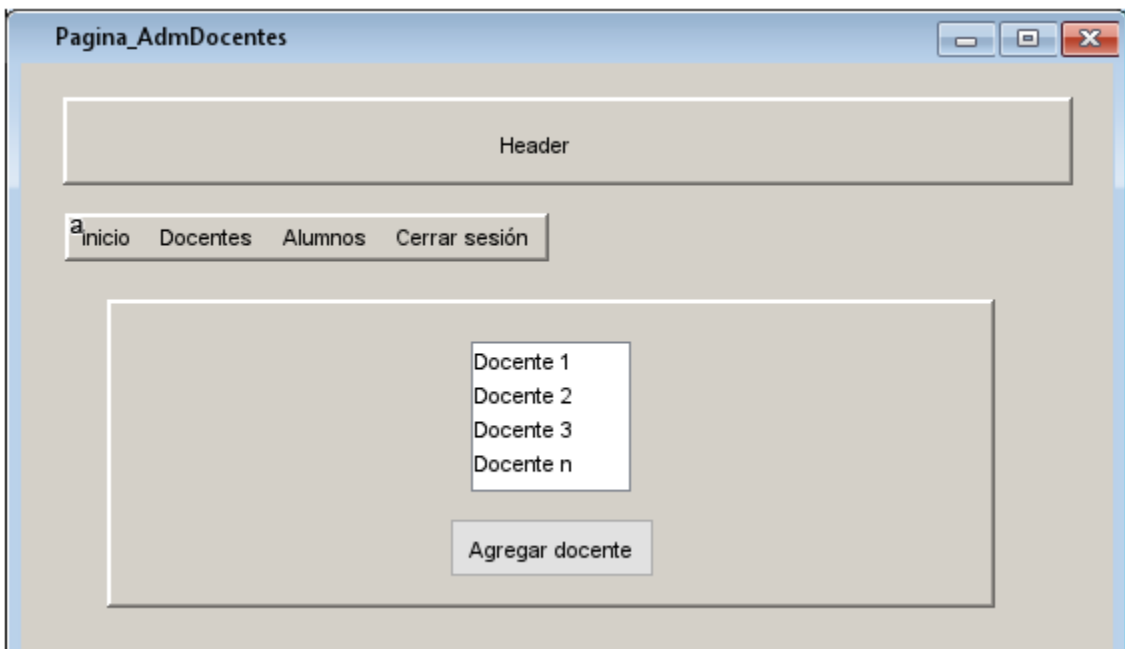


Figura 11 Elaboración propia del prototipo - Pagina Docentes

Si el usuario de tipo administrador desea registrar un docente nuevo en la siguiente imagen (figura 12) se muestra el formulario donde debe ingresar la información del docente. Al finalizar la captura de datos debe dar clic en el botón 'Insertar' para almacenar la información en la base de datos.

The image shows a web browser window with the title "Pagina\_AdmAgregarDocentes". At the top, there is a header area labeled "Header". Below the header is a navigation menu with the following items: "Inicio", "Docentes", "Alumnos", and "Cerrar sesión". The "Alumnos" item is currently selected. Below the navigation menu is a form titled "Capturar datos". The form contains five input fields: "Id Docente:", "Nombre:", "Apellidos:", "Correo:", and "Contraseña:". The "Contraseña:" field is masked with asterisks. Below the input fields is a button labeled "Insertar".

*Figura 12 Elaboración propia del prototipo - Agregar docente*

La interfaz de la opción 'Alumnos' será similar al prototipo de la figura 11, en este caso la lista será consultada de la tabla alumnos. Así mismo, el usuario podrá registrar alumnos nuevos y el diseño de esta interfaz será similar a la figura 12, el sistema solicitará la información a registrar mediante un formulario.

#### 3.5.4 Diseño Interfaz de usuario – Docente

Si el usuario inicia sesión como docente se mostrará la página de inicio (figura 13). El docente podrá navegar en un menú con las opciones: Inicio, Lista alumnos, Retroalimentación y Cerrar sesión. Donde la última opción, claramente es para salir del sistema.

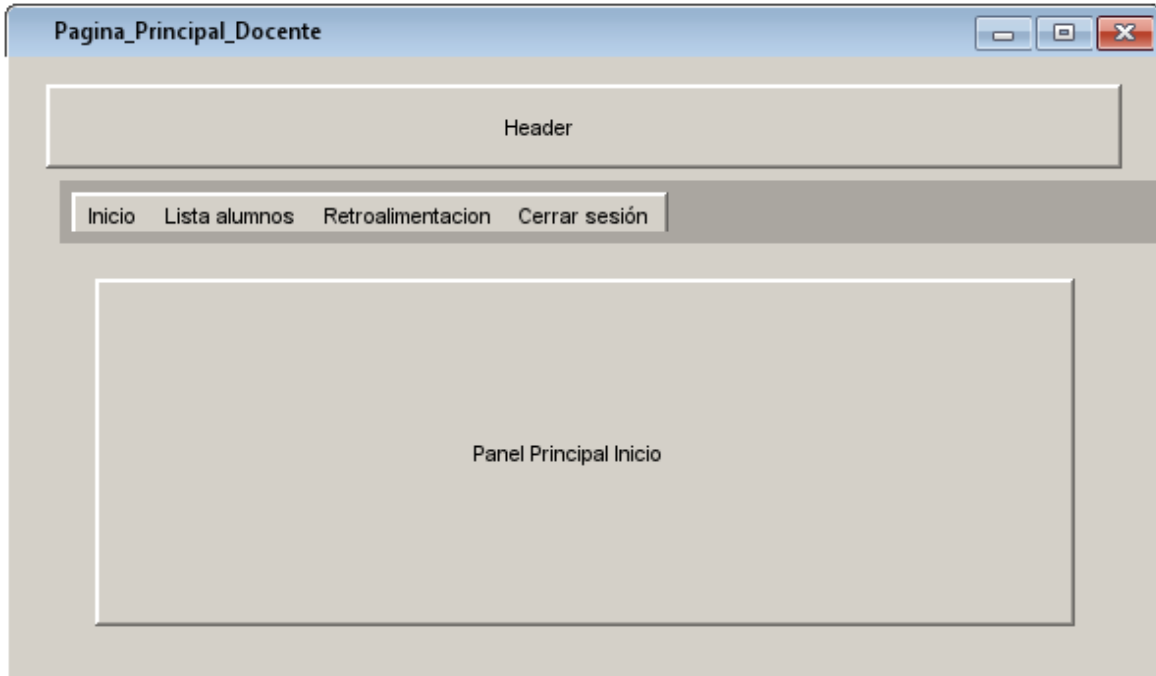


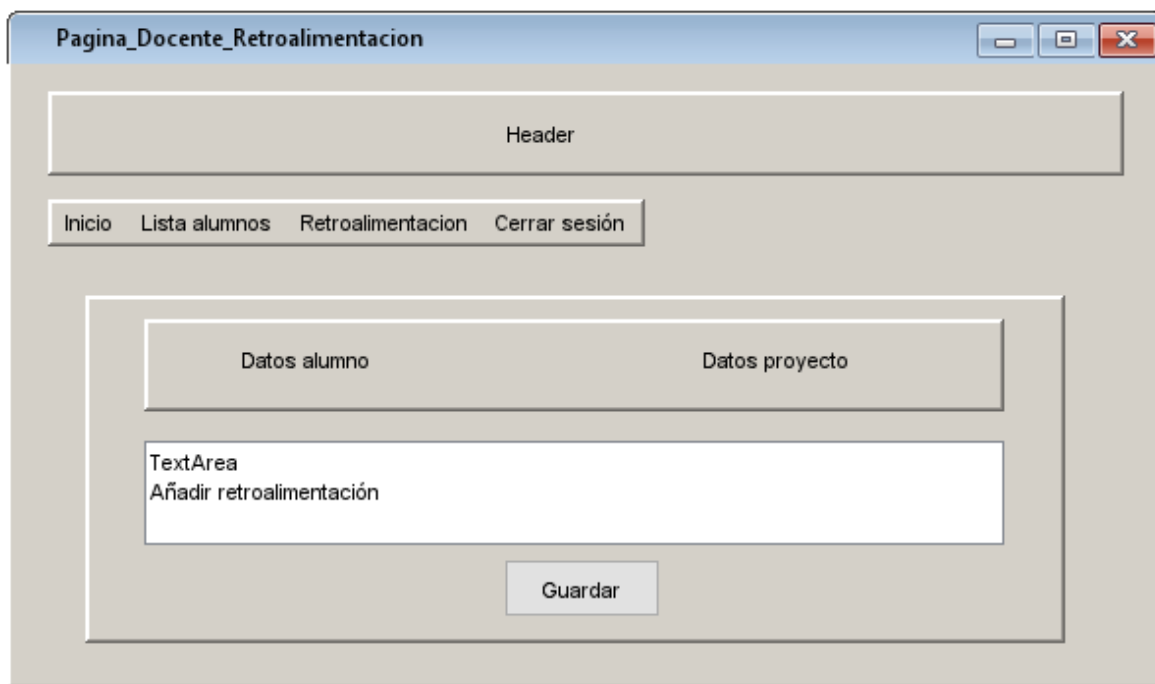
Figura 13 Elaboración propia del prototipo - Pagina Inicio docente

En la figura 14 se muestra la vista que tendrá el docente al seleccionar la opción 'Lista alumnos' donde se enlistaran los alumnos que le fueron asignados.



Figura 14 Elaboración propia del prototipo - Lista de alumnos

La figura 15 es un diseño de la interfaz para la opción 'Retroalimentación' donde el docente podrá agregar sus comentarios en un cuadro de texto.



*Figura 15 Elaboración propia del prototipo – Retroalimentación*

### 3.5.5 Diseño Interfaz de usuario – Alumno

Si el usuario inicia sesión como alumno se mostrará la página de inicio (figura 16). El alumno podrá navegar en un menú con las opciones: Inicio, Datos generales, agregar avances y Cerrar sesión. Donde la última opción, claramente es para salir del sistema.

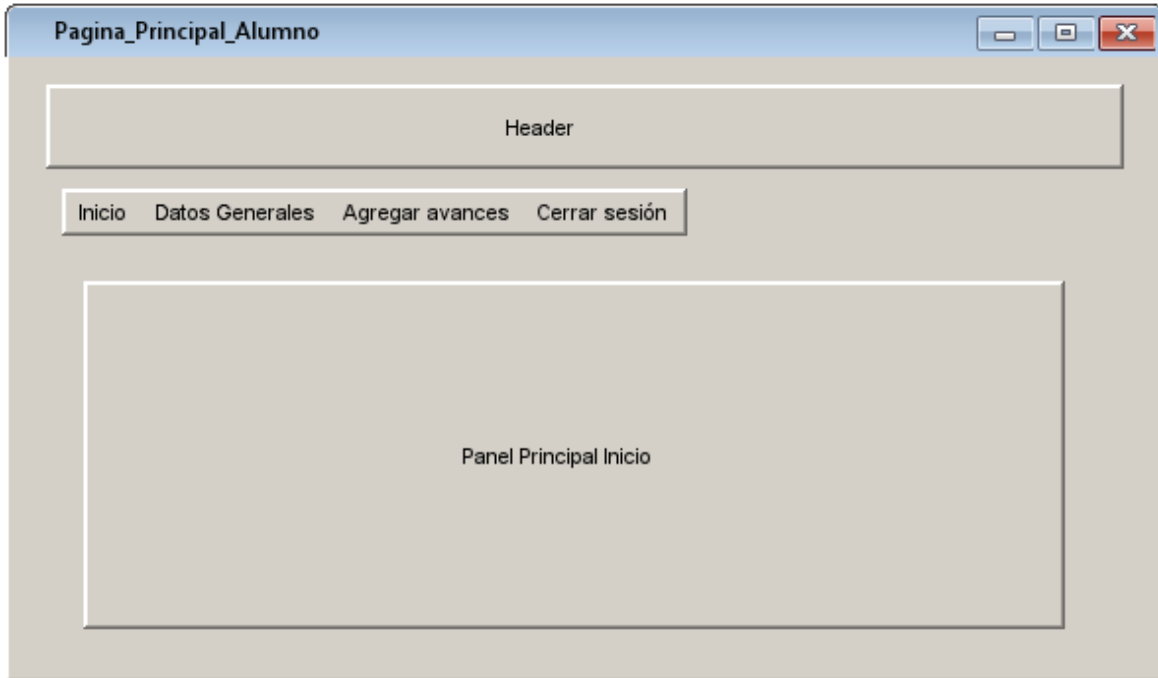


Figura 16 Elaboración propia del prototipo - Pagina inicio Alumno

En la figura 17 se muestra el prototipo de interfaz para la opción 'Datos generales', donde el alumno podrá visualizar su información personal, así como la información correspondiente a su proyecto de tesina.

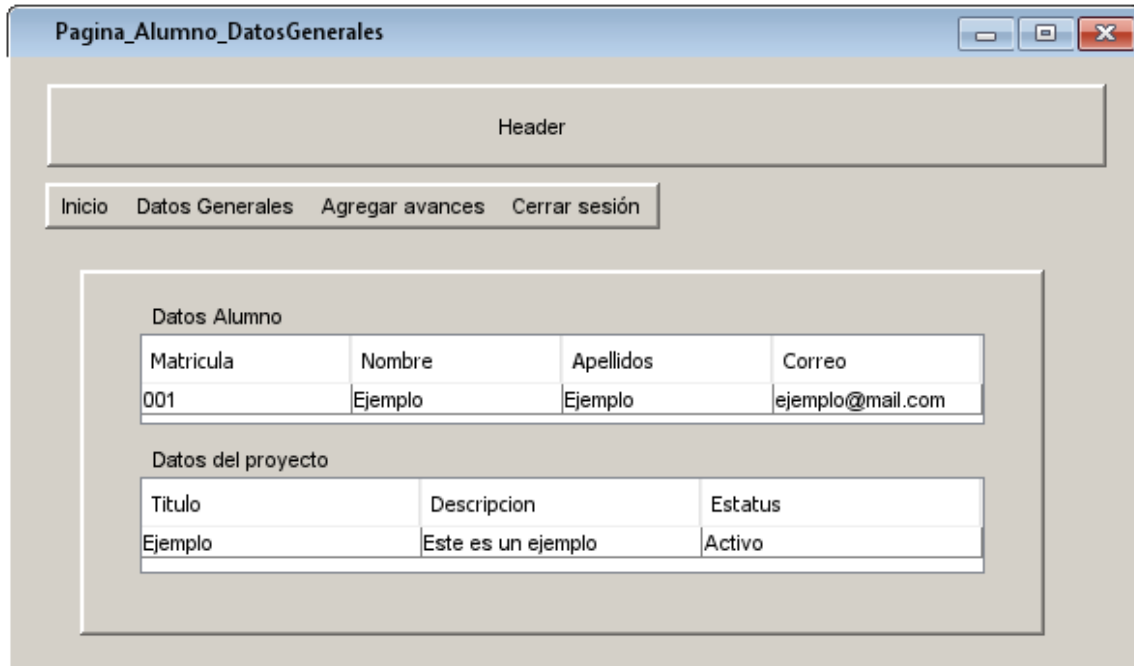


Figura 17 Elaboración propia del prototipo - Datos generales alumno

En la figura 18 se muestra el prototipo de la interfaz para la opción 'Agregar avances', donde el alumno podrá actualizar el estatus de su proyecto, así como la versión de su documento de tesina, para actualizar los datos en la base de datos debe dar clic en el botón 'Actualizar'.

El prototipo de la interfaz de usuario se muestra en una ventana con el título "Pagina\_Alumno\_AgregarAvances". La ventana contiene un encabezado con el texto "Header". Debajo del encabezado hay una barra de navegación con los botones "Inicio", "Datos Generales", "Agregar Avances" (seleccionado) y "Cerrar sesión". El área principal de la interfaz contiene un formulario con los siguientes campos:

- Título:
- Descripción:
- Status:
- Documento:

Debajo del formulario hay un botón "Actualizar".

*Figura 18 Elaboración propia del prototipo - Agregar avances*

### 3.6 Diseño de la Base de Datos

El modelo de datos son abstracciones que permiten la implementación de un sistema eficiente de base de datos, el diseño conceptual es utilizado para representar la realidad a un alto nivel de abstracción [8].

Para el análisis y definición de la estructura de datos vamos a utilizar el modelo relacional, identificando así cada entidad, atributos, relaciones, llaves primarias y foráneas según correspondan.

En la figura 19 se muestra el modelo entidad relación diseñado para el proyecto sistema de gestión de alumnos de titulación.

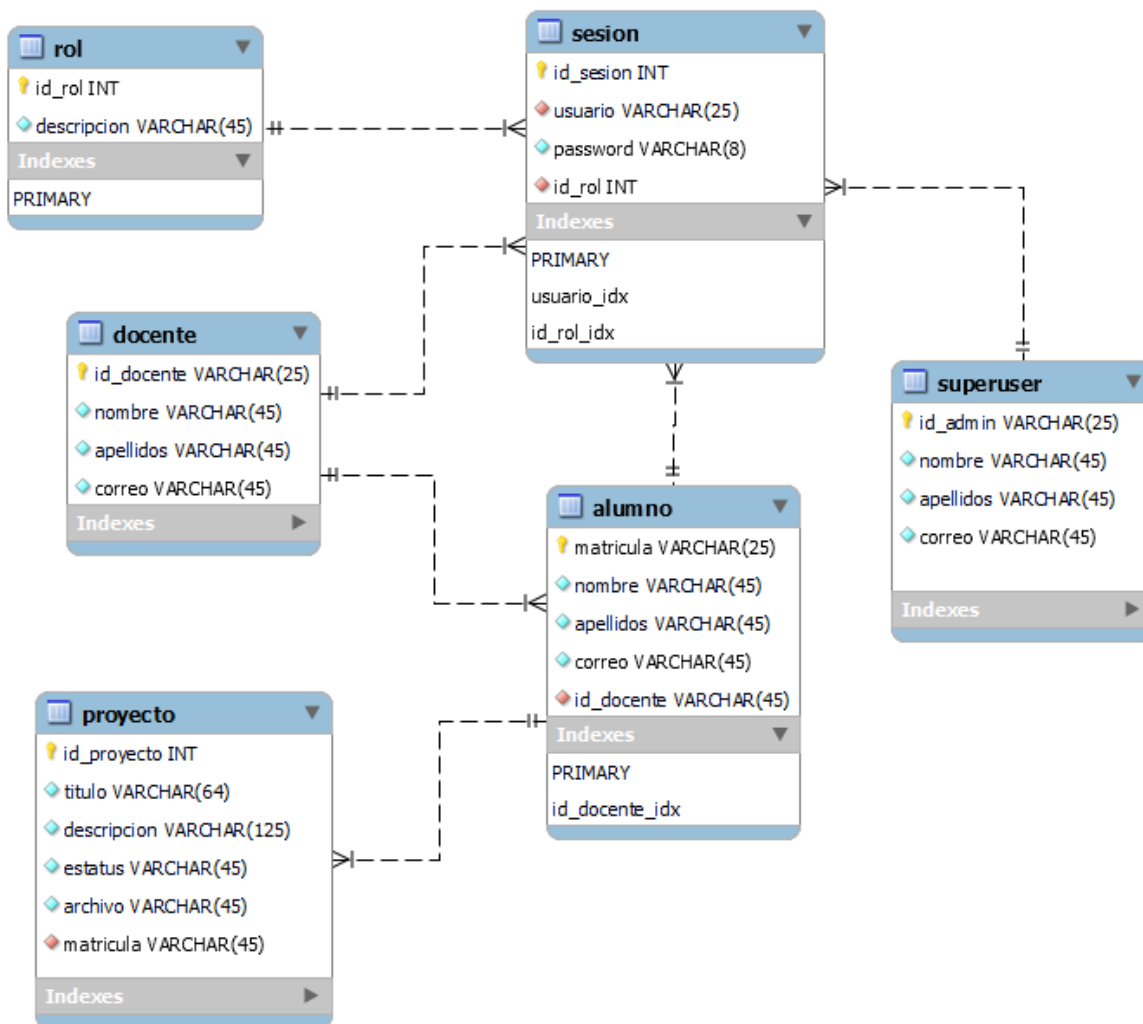


Figura 19 Modelo Entidad - Relación

## CAPITULO 4. Implementación y pruebas del sistema

De acuerdo con los modelos y diagramas obtenidos en las fases de análisis y diseño a continuación se describe el proceso de implementación del presente proyecto.

Así mismo, se incluyen las pruebas realizadas al sistema con el objetivo de garantizar su correcto funcionamiento.

### 4.1 Plataforma de desarrollo.

WampSever es un entorno de desarrollo web, actúa como un servidor virtual en tu computadora sin necesidad de alojarlo en la web por lo que, puedes implementar tu sitio web de manera local.

Para la implementación del sistema para la gestión de alumnos de tesis se hizo uso de WampServer, integrando los lenguajes HTML, PHP y CSS para el desarrollo del sitio y MySQL como gestor de base de datos.

### 4.2 Base de datos.

En base al diseño lógico de la base de datos implementamos nuestro modelo físico mediante el desarrollo de scripts de instrucciones MySQL.

En primera instancia se crea la base de datos utilizando la sentencia:

```
create database tesis;
```

- Entidad superuser

Se crea la estructura para la tabla superuser, en donde se almacenarán los datos de los usuarios de tipo administrador. El campo `id_admin` corresponde a la llave primaria, la cual también será el nombre de usuario para el acceso al sistema.

```
create table superuser(
    id_admin varchar(25) not null primary key,
    nombre varchar(64) not null,
    apellidos varchar(64) not null,
    correo varchar(64) not null
);
```

En la figura 20 se muestra la tabla superuser insertada en la base de datos tesis.

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra	Acción
<input type="checkbox"/>	1 <b>id_admin</b>	varchar(25)	latin1_swedish_ci		No	Ninguna		Cambiar
<input type="checkbox"/>	2 <b>nombre</b>	varchar(64)	latin1_swedish_ci		No	Ninguna		Cambiar
<input type="checkbox"/>	3 <b>apellidos</b>	varchar(64)	latin1_swedish_ci		No	Ninguna		Cambiar
<input type="checkbox"/>	4 <b>correo</b>	varchar(64)	latin1_swedish_ci		No	Ninguna		Cambiar

Figura 20 Estructura entidad superuser

- Entidad docente

Se crea la estructura para la tabla docente, donde se almacenarán los datos del docente. El campo id\_docente corresponde a la llave primaria, el cual también será el nombre de usuario para el acceso al sistema.

```
create table docente(
    id_docente varchar(25) not null primary key,
    nombre varchar(64) not null,
    apellidos varchar(64) not null,
    correo varchar(64) not null
);
```

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra	Acción
<input type="checkbox"/>	1 <b>id_docente</b>	varchar(25)	latin1_swedish_ci		No	Ninguna		Cambiar
<input type="checkbox"/>	2 <b>nombre</b>	varchar(64)	latin1_swedish_ci		No	Ninguna		Cambiar
<input type="checkbox"/>	3 <b>apellidos</b>	varchar(64)	latin1_swedish_ci		No	Ninguna		Cambiar
<input type="checkbox"/>	4 <b>correo</b>	varchar(64)	latin1_swedish_ci		No	Ninguna		Cambiar

Figura 21 Estructura entidad docente

- Entidad alumno

Se crea la estructura para la tabla alumno, donde se almacenarán los datos del alumno. El campo matricula corresponde a la llave primaria, el cual también será su nombre de usuario para acceder al sistema. Esta tabla estará relacionada con la tabla docente mediante la llave foránea id\_docente para poder saber cuántos alumnos tiene asignados cada docente.

```
create table alumno(
    matricula varchar(25) not null primary key,
    nombre varchar(64) not null,
    apellidos varchar(64) not null,
    correo varchar(64) not null,
    id_docente varchar(25) not null,
    foreign key (id_docente) references
    docente(id_docente)
);
```

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra	Acción
<input type="checkbox"/>	1 <b>matricula</b>	varchar(25)	latin1_swedish_ci		No	Ninguna		Cambiar
<input type="checkbox"/>	2 <b>nombre</b>	varchar(64)	latin1_swedish_ci		No	Ninguna		Cambiar
<input type="checkbox"/>	3 <b>apellidos</b>	varchar(64)	latin1_swedish_ci		No	Ninguna		Cambiar
<input type="checkbox"/>	4 <b>correo</b>	varchar(64)	latin1_swedish_ci		No	Ninguna		Cambiar
<input type="checkbox"/>	5 <b>id_docente</b>	varchar(25)	latin1_swedish_ci		No	Ninguna		Cambiar

Figura 22 Estructura entidad alumno

- Entidad rol

Se crea la estructura para la tabla rol, en la cual existirán únicamente tres registros, los cuales son, administrador, docente y alumno. Con esta tabla se clasificarán los usuarios controlando así las interfaces graficas para poder tener un control de la información, creando un nivel de seguridad que solo el usuario correspondiente a los datos los podrá consultar, editar o eliminar según los permisos asignados.

```
create table rol(
```

```

id_rol int not null primary key,
descripcion varchar(64)
);
insert into rol (id_rol, descripcion) values ('1', 'administrador');
insert into rol (id_rol, descripcion) values ('2', 'docente');
insert into rol (id_rol, descripcion) values ('3', 'alumno');

```

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra	Acción
<input type="checkbox"/>	1 <b>id_rol</b>	int(11)			No	Ninguna		Cambiar
<input type="checkbox"/>	2 <b>descripcion</b>	varchar(64)	latin1_swedish_ci		Sí	NULL		Cambiar

Figura 23 Estructura entidad rol

- Entidad sesión

Se crea la estructura para la tabla sesión, en esta tabla el identificador único o llave primaria es el campo id\_sesion. En esta tabla se almacenarán los datos para el inicio de sesión para cada usuario.

El campo usuario serán las llaves foráneas que se obtienen de las tablas: superuser, docente y alumno; para poder obtener su identificar único y asignarlo como nombre de usuario para acceder al sistema de gestión de alumnos de titulación. Así mismo, el campo id\_rol es una llave foránea con la cual identificamos el tipo de usuario para poder asignar la interfaz de usuario que corresponda.

```

create table sesion(
    id_sesion int not null auto_increment primary key,
    usuario varchar(25) not null,
    password varchar(8) not null,
    id_rol int not null,
    foreign key (usuario) references superuser(id_admin),
    foreign key (usuario) references docente(id_docente),
    foreign key (usuario) references alumno(matricula),
    foreign key (id_rol) references rol(id_rol)
)

```

);



#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra	Acción
<input type="checkbox"/>	1 <b>id_sesion</b>	int(11)			No	Ninguna	AUTO_INCREMENT	Cambiar
<input type="checkbox"/>	2 <b>usuario</b>	varchar(25)	latin1_swedish_ci		No	Ninguna		Cambiar
<input type="checkbox"/>	3 <b>password</b>	varchar(8)	latin1_swedish_ci		No	Ninguna		Cambiar
<input type="checkbox"/>	4 <b>id_rol</b>	int(11)			No	Ninguna		Cambiar

Figura 24 Estructura entidad sesión

- Entidad proyecto

Se crea la estructura para la tabla proyecto, en la cual se almacenarán los datos relacionados al proyecto. El campo id\_proyecto corresponde a la llave primaria y para poder saber a qué alumno corresponde el proyecto se crea la llave foránea matricula, con la cual se relacionan las tablas alumno y proyecto.

Esta información únicamente la puede editar el alumno autor del proyecto por lo que es importante el uso de sesiones para asegurar la información por usuario de tipo alumno.

```
create table proyecto(  
    id_proyecto int not null auto_increment primary key,  
    titulo varchar(64) not null,  
    descripcion varchar(128),  
    status varchar(64) not null,  
    archivo varchar(64) not null,  
    matricula varchar(25) not null,  
    foreign key (matricula) references alumno(matricula)  
);
```

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra	Acción
<input type="checkbox"/> 1	<b>id_proyecto</b>	int(11)			No	Ninguna	AUTO_INCREMENT	Cambi
<input type="checkbox"/> 2	<b>titulo</b>	varchar(64)	latin1_swedish_ci		No	Ninguna		Cambi
<input type="checkbox"/> 3	<b>descripcion</b>	varchar(128)	latin1_swedish_ci		Sí	NULL		Cambi
<input type="checkbox"/> 4	<b>status</b>	varchar(64)	latin1_swedish_ci		No	Ninguna		Cambi
<input type="checkbox"/> 5	<b>archivo</b>	varchar(64)	latin1_swedish_ci		No	Ninguna		Cambi
<input type="checkbox"/> 6	<b>matricula</b>	varchar(25)	latin1_swedish_ci		No	Ninguna		Cambi

Figura 25 Estructura entidad proyecto

## 4.3 Interfaz del Sistema

### 4.3.1 Autenticación para acceder al sistema

Para hacer uso del sistema, así como para acceder a la información almacenada en la base de datos es necesario identificarse para definir el tipo de usuario y las funciones con las que podrá interactuar dentro del sistema.

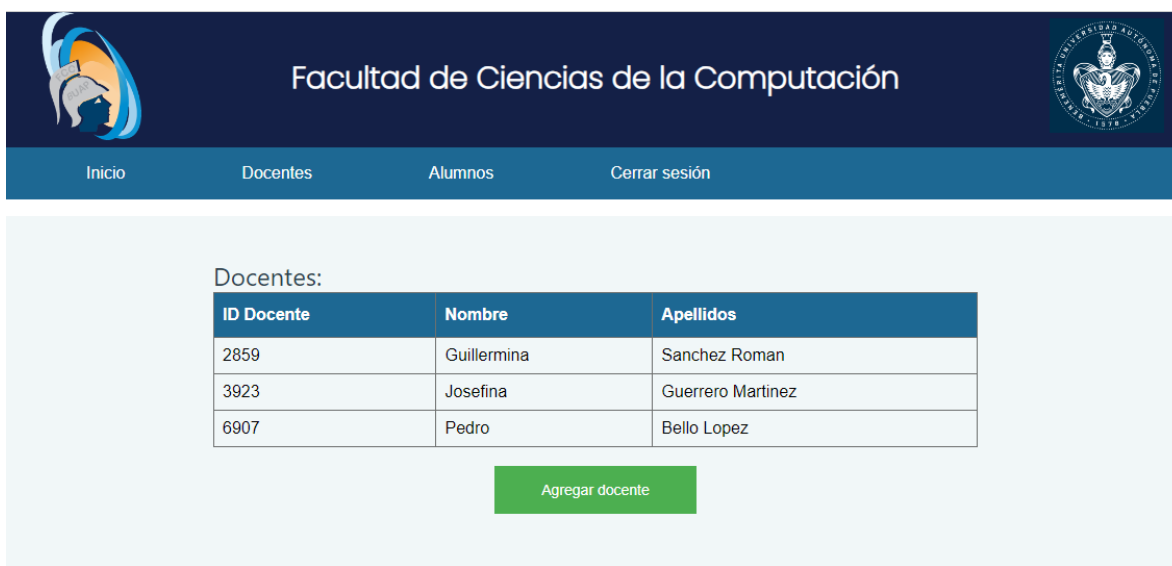
En la figura 26 se muestra la pantalla de acceso, donde se le solicitan al usuario su nombre de usuario y contraseña.

Figura 26 Pantalla inicial - Autenticación de usuario

Los datos ingresados son sometidos a una validación para comprobar su existencia en la base de datos y si estos fueron capturados de manera correcta de lo contrario el sistema le informara por medio de un mensaje el motivo por el cual se le denegó el acceso al sistema.

#### 4.3.2 Interfaz usuario Administrador.

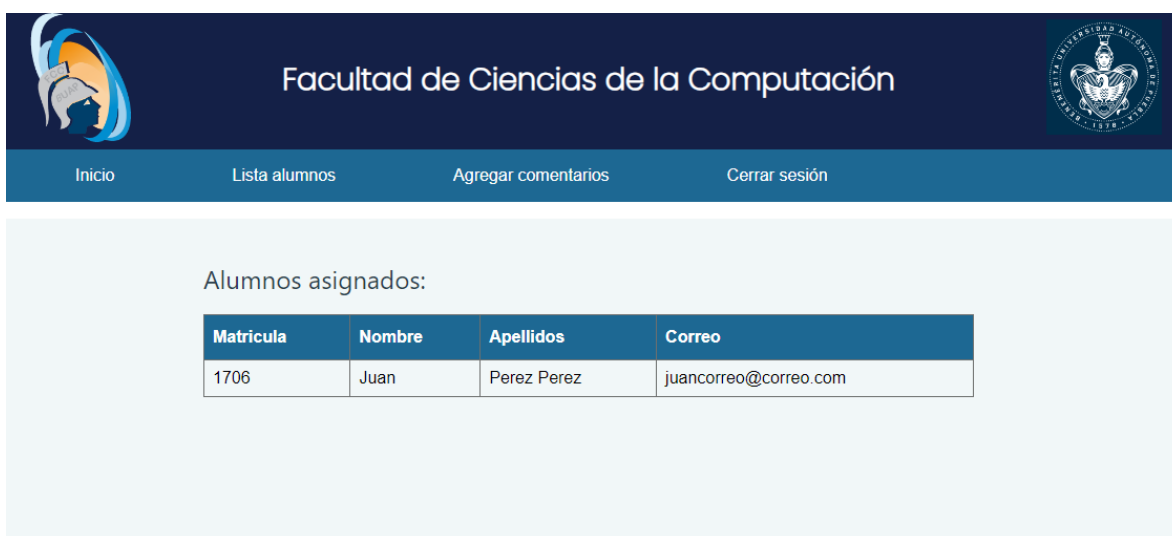
Tal como lo traza el prototipo se genera la pantalla para el usuario de tipo administrador, agregado un menú principal el cual se muestra en la siguiente figura.



*Figura 27 Pantalla principal Administrador*

#### 4.3.3 Interfaz usuario Docente.

Del mismo modo, en la interfaz para el usuario de tipo docente se agrega el menú con las funcionalidades correspondientes a un usuario docente.

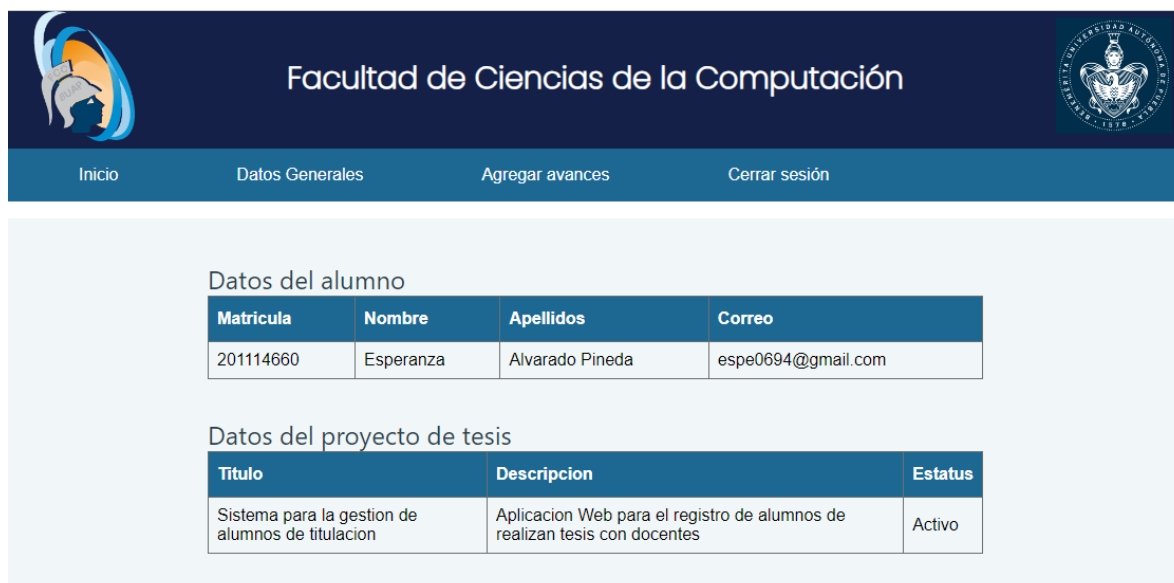


*Figura 28 Pantalla principal Docente*

#### 4.3.4 Interfaz usuario Alumno

En la figura 29 se muestra la pantalla para el usuario autenticado como alumno. Solo se muestra la información correspondiente al alumno que inicio sesión, por lo que es importante que los usuarios usen una contraseña segura y que no sea compartida ya que esto puede generar ataques a la base de datos.

Cada alumno podrá ir actualizando el estatus de su proyecto para que de esta manera su asesor asignado este enterado de la situación de cada alumno para realizar el seguimiento u observaciones que correspondan.



Facultad de Ciencias de la Computación

Inicio Datos Generales Agregar avances Cerrar sesión

Datos del alumno

Matricula	Nombre	Apellidos	Correo
201114660	Esperanza	Alvarado Pineda	espe0694@gmail.com

Datos del proyecto de tesis

Título	Descripción	Estatus
Sistema para la gestión de alumnos de titulación	Aplicación Web para el registro de alumnos de realización de tesis con docentes	Activo

Figura 29 Pantalla principal Alumno

#### 4.4 Pruebas

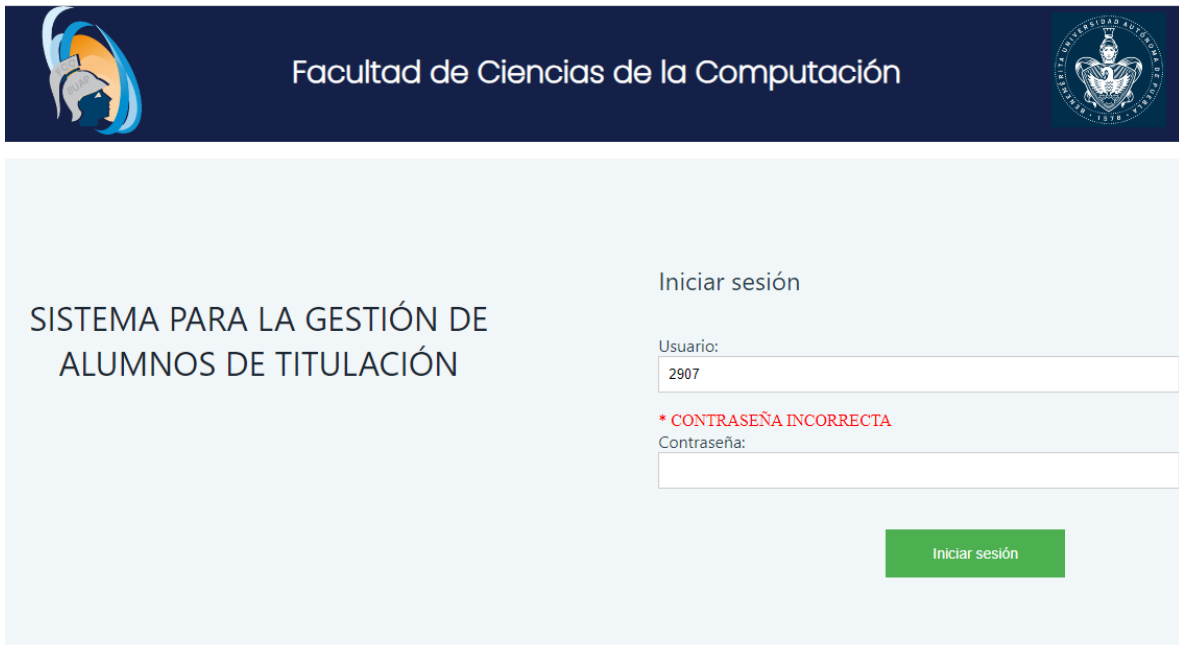
La fase de pruebas dentro de un proyecto de software es fundamental para evaluar y verificar los resultados del proyecto por lo que en el presente proyecto se han aplicado una serie de pruebas para garantizar el funcionamiento especificado en los requerimientos.

A continuación, se describen las pruebas de seguridad, rendimiento y de unidad realizadas para los tres diferentes tipos de usuarios del sistema de gestión de alumnos de titulación.

- Acceso al sistema

Es importante garantizar el funcionamiento del módulo de autenticación del usuario, ya que si este es incorrecto los usuarios no podrán acceder a su información o podrían acceder a funcionalidades que no les corresponden etc.

Por lo que en la figura 30 se muestra un ejemplo de validación. El usuario intenta iniciar sesión con una contraseña incorrecta, por lo que el sistema muestra un mensaje informativo que el dato que proporciono es incorrecto. Así mismo, el sistema informa si el nombre de usuario es incorrecto o no existe en la base de datos.



The image shows a web interface for a student management system. At the top, there is a dark blue header with a logo on the left and the text 'Facultad de Ciencias de la Computación' in the center, and a circular seal on the right. Below the header, the main content area is light blue. On the left, the text 'SISTEMA PARA LA GESTIÓN DE ALUMNOS DE TITULACIÓN' is displayed. On the right, there is a login section titled 'Iniciar sesión'. It contains two input fields: 'Usuario:' with the value '2907' and 'Contraseña:'. Below the password field, a red error message reads '\* CONTRASEÑA INCORRECTA'. At the bottom right of the login section, there is a green button labeled 'Iniciar sesión'.

*Figura 30 Prueba validación de usuario*

- Usuario administrador

Una de principales funciones como usuario de tipo administrador es registrar a demás usuarios, por lo que el sistema debe garantizar el rendimiento para almacenar todos los registros necesarios además de proteger la información de todo tipo de vulnerabilidades. En la figura 31 se muestra el formulario para el registro de un nuevo usuario de tipo docente y en la figura 32 el formulario para el registro de un nuevo usuario de tipo alumno.

Facultad de Ciencias de la Computación

Inicio Docentes Alumnos Cerrar sesión

Capturar datos:

ID Docente:

Nombre:

Apellidos:

Correo:

Password:

Insertar

Figura 31 Pantalla formulario nuevo docente

Facultad de Ciencias de la Computación

Inicio Docentes Alumnos Cerrar sesión

Capturar datos:

Matricula:

Nombre:

Apellidos:

Correo:

Asesor:  | Contraseña:

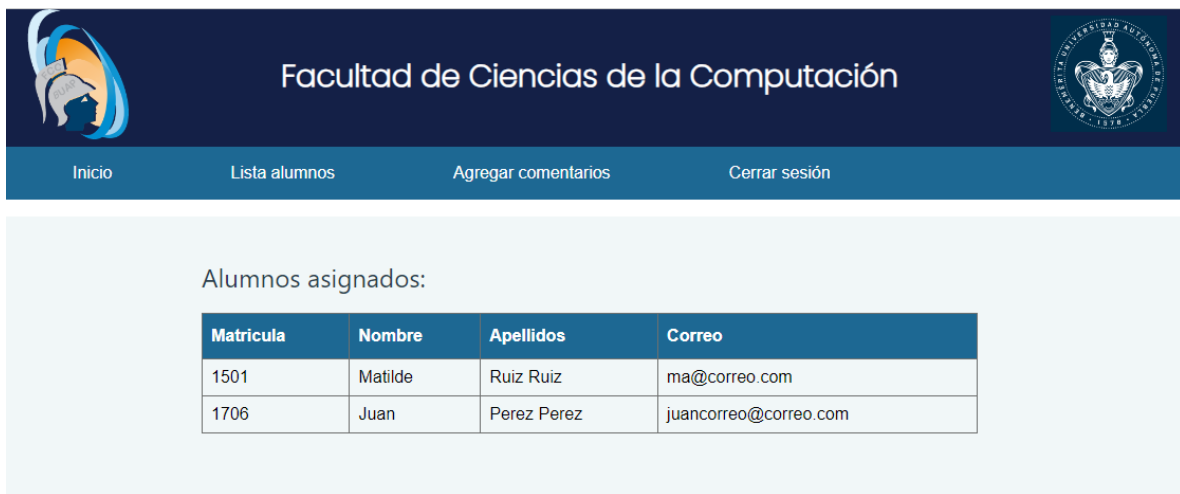
Insertar

Figura 32 Pantalla formulario nuevo alumno

- Usuario docente

Los usuarios autenticados como docentes tendrán permiso para visualizar la lista de alumnos asignados, así como sus datos generales, por lo que el tipo de prueba a realizar es de rendimiento y seguridad al limitar el acceso a la información correspondiente: el sistema debe realizar la consulta en el menor tiempo posible.

En la figura 33 se muestra la pantalla de la lista de alumnos por docente que accedió al sistema.



The screenshot shows a web interface for the Faculty of Computer Sciences. At the top, there is a dark blue header with the faculty name and a navigation bar with buttons for 'Inicio', 'Lista alumnos', 'Agregar comentarios', and 'Cerrar sesión'. Below the header, the text 'Alumnos asignados:' is displayed above a table with two rows of student data.

Matricula	Nombre	Apellidos	Correo
1501	Matilde	Ruiz Ruiz	ma@correo.com
1706	Juan	Perez Perez	juancorreo@correo.com

*Figura 33 Pantalla lista de alumnos*

- Usuario alumno

Los alumnos deben actualizar el estatus de su proyecto, así como adjuntar el documento de su tesina actualizado. El documento para adjuntar debe ser de tipo PDF.

En la figura 34 se muestra el formulario para que el alumno actualice los datos de su proyecto.



Título :

Descripcion :

Status :

Documento:  Ninguno archivo selec.

Actualizar

*Figura 34 Pantalla formulario actualizar estatus*

## Conclusión

Hoy en día, gracias al internet el desarrollo de aplicaciones web como soluciones a diferentes necesidades ha creado un gran impacto, ya que estas herramientas facilitan las tareas, procesos lógicos u operativos. Además de que un entorno web es accesible para todo tipo de dispositivo en el que sea abierto un navegador web. En el ámbito educativo estas herramientas son imprescindibles para administrar, gestionar y realizar procesos en línea.

A pesar de que en el mercado ya existen aplicaciones para la gestión de alumnos, en este proyecto se cumple con el objetivo principal al desarrollar una aplicación web para la gestión de alumnos de titulación por medio de tesis, diseñado y elaborado para la Facultad de Ciencias de la Computación de la Benemérita Universidad Autónoma de Puebla, contemplando todos los requerimientos descritos en la fase inicial del proyecto.

Así mismo se cumple con el objetivo de facilitar al docente la administración y seguimiento de los alumnos que están en proceso de titulación. Por otra parte, esta herramienta es considerada como otro medio de comunicación entre alumnos y docentes ya que con el uso de esta herramienta ambas partes estarán en constante interacción.

Por último, se cumple con el objetivo de diseñar y crear un prototipo de software siguiendo la metodología RUP, el sistema para la gestión de alumnos de titulación fue desarrollado con PHP, HTML y CSS y la base de datos, así como las sentencias en MySQL.

## Trabajo a futuro

Es importante mencionar que el presente prototipo ya es funcional y cumple con los objetivos principales, sin embargo, se pueden mejorar y/o agregar funcionalidades para obtener una herramienta más completa. Algunas de las mejoras son:

- Enviar un correo electrónico cuando el docente agregue un comentario al proyecto. De momento si el docente agrega una retroalimentación el alumno solo la podrá ver cuando acceda al sistema, por lo que una mejora es enviar un correo al alumno notificando el contenido de la retroalimentación.
- Renombrar los nombres de las tablas de la base de datos. De esta manera se puede prevenir un ataque ya que de momento los nombres son muy predecibles poniendo el riesgo la información de los usuarios.
- Crear un respaldo de la base de datos. Hacer una copia de la base de datos en caso de que sea necesaria una recuperación operativa.
- Colaboración con otras facultades. Se espera que el sistema para la gestión de alumnos de titulación tenga un efecto positivo en los docentes lo cual permita que sea distribuido en otras facultades.

## Bibliografía

1. Philippe Kruchten, The Rational Unified Process An Introduction, Addison Wesley, 2001.
2. Booch, Grady / Rumbaugh, James / Jacobson, Ivar. E Proceso Unificado De Desarrollo De Software. Addison Wesley, 2000
3. PIÑEIRO GOMEZ, J. M. (2022). Entornos de desarrollo. España: Ediciones Paraninfo, S.A.
4. IAN SOMMERVILLE (2005), Ingeniería de Software Séptima edición, Pearson Educación, S.A.
5. Pressman, R. S. (2002). INGENIERIA DE SOFTWARE. España: McGraw-Hill Interamericana de España S.L.
6. Ullman, L. (2009). PHP paso a paso. España: ANAYA MULTIMEDIA.
7. Cobo, Á. (2005). PHP y MySQL: Tecnología para el desarrollo de aplicaciones web.. España: Editorial Díaz de Santos, S.A..
8. Welling, L., Thomson, L. (2009). Desarrollo Web con PHP y MySQL. España: ANAYA MULTIMEDIA.
9. Catherine M. Ricardo (2009). Bases de datos: MCGRAW-HILL INTERAMERICANA EDITORES, S.A. de C.V.
10. Elizabeth Freeman & Erick Freeman. Head First HTML with CSS & XHTML–2006.
11. Carlos Coronel, Steven Morris, Peter Rob. Bases de datos: diseño, implementación y administración, México 2002.
12. Silberschatz, Korth, Sudarshan, Fundamentos de Diseño de Bases de Datos, quinta edición, Mc Graw Hill, España, 2007.

## Mesografía

13. Classter  
<https://www.classter.com/>
14. Algebraix  
<https://www.algebraix.com>
15. EL LENGUAJE UNIFICADO DE MODELADO (UML)  
<http://profesores.fi-b.unam.mx/carlos/aydoo/uml.html>
16. Modelo cliente servidor  
<https://blog.infranetworking.com/modelo-cliente-servidor/>
17. Introducción al CSS: qué es, para qué sirve y otras 10 preguntas frecuentes  
<https://blog.hubspot.es/website/que-es-css>
18. Sistema Gestor de Base de Datos  
[https://www.ecured.cu/Sistema\\_Gestor\\_de\\_Base\\_de\\_Datos](https://www.ecured.cu/Sistema_Gestor_de_Base_de_Datos)